* Adoption of a secure coding standard, and not leaving security to the end

By adopting a secure coding standard, a development team has a more clear idea of what should be tested, what areas need improvement once the final product is released, and how they should design their system to accommodate security. The final point is important and should not be overlooked, as designing a system with security in mind means the system will have a much lower chance of compounding vulnerabilities. If security is left until the end of the process, components of the system may have been built upon vulnerable components, which sometimes leads to those components being vulnerable themselves. This could mean the entire system has to be changed to fix vulnerabilities that would have appeared early in the development process, taking up valuable time and assets to fix.

* Evaluation and assessment of risk and cost benefit of mitigation

As systems become more complex in their design, vulnerabilities become increasingly more likely and nearly impossible to fully avoid. Instead of just focusing on fixing specific vulnerabilities, developers should also use mitigation strategies to lower the severity of attacks. If an attacker hacks into a server, for instance, this shouldn’t give them access to the entire network, as defense-in-depth strategies should have been used to mitigate this issue. If an attacker hacks into a database, any private information should be encrypted to mitigate the threat of stealing data. By not putting all the eggs in one basket, the cost of fixing each issue if any attack occurs becomes cheaper.

* Zero trust

Regardless of where a user signs into a network, that user will still need to be verified as though they’re signing in from an open network. This is the concept of zero trust. Even if the origin of the sign in attempt comes from within the companies building using a company computer, the network will treat this with no more leniency than a user signing in from a public library. This principle of security ensures that even if an attacker bypassed the network's firewall or found their way into the company's building, they would still need proper verification to enter the network.

* Implementation and recommendations of security policies

Before development even starts, security policies should be clearly established. As mentioned previously, security should not be saved for the end and should be used throughout the development process, meaning it’s important to establish the standards of security that must be implemented. One important policy that should be utilized in almost all systems would be defense-in-depth (DID). DID breaks the systems security up into layers, where each layer has their own security procedures. If one of these layers is broken, the other layers don’t become compromised. The Triple A policy should also be used for nearly all systems, as proper authentication, authorization, and accounting procedures ensure the actions of users are restricted and monitored, lowering the chance of verification based attacks.